CLAIMS

1. A method for modeling an information request, comprising:

receiving unlabeled and labeled documents;

extracting a set of features from each document;

learning a model from example documents marked as positive or negative with respect to said request wherein said model scores documents to evaluate a degree of membership in a group responsive to said request;

evaluating the performance of model settings on example documents;

applying an adjustment algorithm that provides a threshold value θ_{new};

applying a scoring function that computes *score*, a value assigned to a document by the learnt model, and classifies the document based on the sign of the following equation:

$$Class(X) = Sign(score - \theta_{new})$$

2. The method according to claim 1, wherein:

said model is a support vector machine determined by training data from labeled example documents.

3. The method according to claim 1, wherein:

said model comprises a list of terms and weights extracted from labeled example documents.

4. The method according to claim 1, wherein:

said model comprises a list of terms and corresponding weights and a threshold value determined by:

extracting terms and features from the positive and negative documents;

ranking terms and features;

selecting a subset of terms and features from the ranked terms and features;

assigning a weight w_i for each term and feature;

setting a threshold θ for the model to zero.

5. The method according to claim 4, wherein:

said terms and features are ranked in decreasing order of their Rocchio score calculated as follows:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

in which

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

6. The method according to claim 4, wherein:

said terms and features are assigned a weight as follows:

$w_i = Rocchio_i \cdot idf_i$, calculated as:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

where

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

and idf_i is calculated as follows:

$idf_i = log_2(S/n_i) + 1$

where S is the count of documents in the set and n_i is the count of the documents in which i^{th} feature occurs.

7. A system for filtering documents, comprising:

a computer coupled to a network wherein said computer receives documents over said network and transmits documents to an individual user over said network, wherein said computer:

receiving unlabeled and labeled documents;

extracting a set of features from each document;

learning a model from the example documents marked as positive or negative with respect to a category wherein said model scores documents to evaluate a degree of membership in said category;

evaluating the performance of model settings on example documents;

applying an adjustment algorithm that provides a threshold value θ_{new};

applying a scoring function that computes *score*, a value assigned to a document by the learnt model, and classifies the document based on the sign of the following equation:

$$Class(X) = Sign(score - \theta_{new})$$

8. A system as in claim 7, wherein:

said model is a support vector machine determined by training data from labeled example documents.

9. A system as in claim 7, wherein:

said model comprises a list of terms and weights extracted from labeled example documents.

10. The system of claim 7, wherein:

said model comprises a list of terms and corresponding weights and a threshold value determined by:

extracting terms and features from the positive and negative documents;

ranking terms and features;

selecting a subset of terms and features from the ranked terms and features;

assigning a weight w_i for each term and feature;

setting a threshold θ for the model to zero.

11. The system of claim 10, wherein:

said terms and features are ranked in decreasing order of their Rocchio score calculated as follows:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

in which

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

12. The system of claim 10, wherein:

said terms and features are assigned a weight as follows:

$w_i = Rocchio_i \cdot idf_i$, calculated as:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

where

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

and idf_i is calculated as follows:

$idf_i = log_2(N/n_i) + 1$

where N is the count of documents in the set and n_i is the count of the documents in which i^{th} feature occurs.

13.A method for retrieving information in response to a request, comprising:

receiving unlabeled and labeled documents;

extracting a set of features from each document;

learning a model from example documents marked as positive or negative with respect to said request wherein said model scores documents to evaluate a degree of membership in a group responsive to said request;

evaluating the performance of model settings on example documents;

applying an adjustment algorithm that provides a threshold value θ_{new};

applying a scoring function that computes *score*, a value assigned to a document by the learnt model, and classifies the document based on the sign of the following equation:

$$Class(X) = Sign(score - \theta_{new})$$

14.The method according to claim 13, wherein:

said model is a support vector machine determined by training data from labeled example documents.

15.The method according to claim 13, wherein:

said model comprises a list of terms and weights extracted from labeled example documents.

16.The method according to claim 13, wherein:

said model comprises a list of terms and corresponding weights and a threshold value determined by:

extracting terms and features from the positive and negative documents;

ranking terms and features;

selecting a subset of terms and features from the ranked terms and features;

assigning a weight w_i for each term and feature;

setting a threshold θ for the model to zero.

17. The method according to claim 16, wherein:

said terms and features are ranked in decreasing order of their Rocchio score calculated as follows:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

in which

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

18. The method according to claim 16, wherein:

said terms and features are assigned a weight as follows:

$w_i = Rocchio_i \cdot idf_i$, calculated as:

$$Rocchio_i = TF_{i,Text} + \alpha\left(\frac{1}{R}\sum_{Dj \in D_R \& w_i \in D_i} TF_{ij}\right) - \beta\left(\frac{1}{N}\sum_{Dj \in D_N \& w_i \in D_i} TF_{ij}\right)$$

where

$TF_{i,Text}$: The frequency of feature in the text description of the information need

TF_{ij}: The number of occurences of *feature* in document *j*

D_R: Positive document set

D_N: Negative document set

R: the number of positive documents (i.e., the size of D_R)

N: the number of negative documents

and idf_i is calculated as follows:

$idf_i = log_2(S/n_i) + 1$

where S is the count of documents in the set and n_i is the count of the documents in which i^{th} feature occurs.